IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : C. Brian Atkins
Serial No. : 10/675,823
Filed : Sep. 30, 2003
Title : SINGLE PASS AUTOMATIC PHOTO ALBUM PAGE LAYOUT

Art Unit : 2179
Examiner : Augustine, Nicholas
Confirmation No.: 6652

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

I. Real Party in Interest

The real party in interest is Hewlett-Packard Development Company, L.P., a Texas Limited Partnership having its principal place of business in Houston, Texas.

II. Related Appeals and Interferences

Appellant is not aware of any related appeals or interferences that will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. Status of Claims

Claims 1-21 are pending.

Claims 1-21 stand rejected.

Appellant appeals all rejections of the pending claims 1-21.

CERTIFICATE OF TRANSMISSION

I hereby certify that this document is being transmitted to the Patent and Trademark Office via electronic filing on the date shown below

June 9, 2008
Date of Transmission

(Signature of person mailing papers)

Edouard Garcia
(Typed or printed name of person mailing papers)

IV. Status of Amendments

The amendments filed May 22, 2007, have been entered and acted upon by the Examiner.

No amendments were filed after the final Office action dated February 7, 2008.

V. Summary of Claimed Subject Matter

A. Independent claim 1

The aspect of the invention defined in independent claim 1 is a method for arranging a set of objects within an area (page 24, lines 10-19; FIGS. 11A and 11B). In accordance with this method a first current binary tree comprising a leaf node is initiated (page 24, lines 20-21; FIG. 11A, block 1101). A first object selected from the set is associated with the leaf node (page 24, lines 20-21; FIG. 11A, block 1101). The method comprises establishing candidate binary trees, wherein each of the candidate binary trees comprises the current binary tree and a respective leaf node associated with another object selected from the set, and locations of the leaf nodes within each of the candidate binary trees correspond to relative positions of the associated objects within the area (page 24, line 25 - page 25, line 29; FIG. 11A, blocks 1105-1119). A respective score is computed for each of the candidate binary trees (page 25, lines 7-10; FIG. 11B, block 1110). One of the candidate binary trees is selected as the current binary tree based on the computed scores (page 25, lines 10-22; FIG. 11B, blocks 1110-1111 and 1114-1115). The establishing, the computing, and the selecting are repeated until the current binary tree includes all the objects in the set (page 25, lines 22-29; FIG. 11A, blocks 1116, 1118). After the repeating, the objects are arranged within the area in accordance with the locations of the leaf nodes within the current binary tree (page 25, lines 28-29; page 16, line 10 - page 17, line 2; page 19, lines 7-27; FIG. 6, blocks 604, 605).

B. Claim 3

Claim 3 depends from claim 2 and recites that the establishing of candidate binary trees comprises: removing a subtree of the current binary tree associated with a selected position within the current binary tree (page 25, lines 1-4; FIG. 11A, block 1107); inserting a new interior node into the current binary tree at the selected position (page 25, lines 1-4; FIG.

11A, block 1107); associating either a horizontal or a vertical partition of the area with the new interior node (page 25, lines 4-5; FIG. 11A, block 1107); inserting into the binary tree a new leaf node emanating from the new interior node (page 25, lines 1-4; FIG. 11A, block 1107); associating the new leaf node with the other object selected from the set; and inserting the previously removed subtree back into the binary tree at the new interior node (page 24, lines 26-27; FIG. 11A, block 1105).

C. Claim 4

Claim 4 depends from claim 3 and recites that the selected position is selected from the group consisting of: a leaf node in the current binary tree; and an interior node in the current binary tree (page 24, line 27 - page 25, line 4; FIG. 11A, block 1107).

D. Claim 5

Claim 5 depends from claim 2 and recites further comprising normalizing each of the candidate binary trees (page 25, line 5; FIG. 11A, block 1108), wherein the normalizing comprises: for each of the interior nodes in the candidate binary tree, characterizing a respective bounding box for the objects included in the subtree rooted in the interior node (page 26, lines 25-27; FIG. 14, block 1401); and for each of the objects, allocating a respective region of the area in accordance with the respective bounding box (page 26, lines 27-28; FIG. 14, block 1402).

E. Claim 7

Claim 7 depends from claim 1 and recites that the computing of a respective score for each of the candidate binary trees comprises assessing minimum and maximum object size values for all the objects in the area (page 27, lines 9-10 and 13-14; claim 7 as originally filed). Claim 7 also recites that the selecting of one of the candidate binary trees as the current binary tree based on the computed scores comprises selecting as the current binary tree the candidate binary tree having a greatest respective ratio of minimum area object size value divided by maximum area object size value(FIG. 11B, blocks 1110, 1111, 1114, 1115, and 1118; claim 7 as originally filed).

F. Independent claim 8

The aspect of the invention defined in independent claim 8 is a method for arranging a set of objects within an area(page 24, lines 10-19; FIGS. 11A and 11B). In accordance with this method, a tree structure is established (page 24, lines 20-21; FIG. 11A, block 1101). A first object selected from the set is associated with the tree structure to form a current candidate tree (page 24, lines 20-21; FIG. 11A, block 1101). The current candidate tree is modified to form alternate candidate trees by associating a subsequent object selected from the set with different respective locations on the current candidate tree, wherein the respective locations correspond to relative positions of the associated objects within the area (page 24, line 25 - page 25, line 29; FIG. 11A, blocks 1105-1119). Scores are computed for each of the alternate candidate trees (page 25, lines 7-10; FIG. 11B, block 1110). One of the alternate candidate trees is selected as the current candidate tree based on the computed scores (page 25, lines 10-22; FIG. 11B, blocks 1110-1111 and 1114-1115). The modifying, the computing, and the selecting are repeated until all the objects in the set are associated with the current candidate tree (page 25, lines 22-29; FIG. 11A, blocks 1116, 1118). After the repeating, all the objects in the set are arranged within the area in accordance with the locations in the current candidate tree associated with the objects (page 25, lines 28-29; page 16, line 10 - page 17, line 2; page 19, lines 7-27; FIG. 6, blocks 604, 605).

G. Independent claim 15

The aspect of the invention defined in independent claim 15 is a method for arranging a set of objects within an area (page 24, lines 10-19; FIGS. 11A and 11B). In accordance with this method, a current candidate tree having at least one interior node, at least one leaf node connected to the interior node, and at least one object selected from the set associated with the leaf node is established (page 24, lines 20-21; FIG. 11A, block 1101). The current candidate tree is modified to form alternate candidate trees by associating a subsequent object selected from the set with a different respective location on the current candidate tree, wherein the respective locations correspond to relative positions of the associated objects within the area (page 24, line 25 - page 25, line 29; FIG. 11A, blocks 1105-1119). Scores are computed for each of the alternate candidate trees (page 25, lines 7-10; FIG. 11B, block 1110). One of the alternate candidate trees is selected as the current candidate tree based on

the computed scores (page 25, lines 10-22; FIG. 11B, blocks 1110-1111 and 1114-1115). The modifying, the computing, and the selecting are repeated until all the objects in the set are associated with the current candidate tree (page 25, lines 22-29; FIG. 11A, blocks 1116, 1118). After the repeating, all the objects in the set are arranged within the area in accordance with the locations in the current candidate tree associated with the objects (page 25, lines 28-29; page 16, line 10 - page 17, line 2; page 19, lines 7-27; FIG. 6, blocks 604, 605).

VI. Grounds of Rejection to be Reviewed on Appeal

Claims 1-21 stand rejected under 35 U.S.C. § 102(b) over Geigel (U.S. 2002/0122067).

VII. Argument

A. Applicable standards for sustaining a rejection under 35 U.S.C. § 102(b)

The relevant part of 35 U.S.C. § 102(b) recites that "A person shall be entitled to an invention, unless - ... the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of the application for patent in the United States." Anticipation under 35 U.S.C. § 102(b) requires that each and every element of the claimed invention be present, either expressly or inherently, in a single prior art reference. EMI Group N. Am., Inc., v. Cypress Semiconductor Corp., 268 F.3d 1342, 1350 (Fed. Cir. 2001). Anticipation must be proved by clear and convincing evidence. Electro Medical Systems, S.A. v. Cooper Life Sciences, Inc., 34 F3d 1048, 1052 (Fed. Cir. 1994).

B. Rejection under 35 U.S.C. § 102(b) over Geigel

The Examiner has rejected claims 1-21 under 35 U.S.C. § 102(b) over Geigel (U.S. 2002/0122067).

1. Overview of Geigel's disclosure

Geigel discloses an automatic page layout system that includes a page creator module and an image placement module (see abstract, ¶¶ 54, 59, 64, 77, and FIG. 7).

Geigel's page creator module assigns images in a collection to album pages based on a first genetic evolution algorithm (see abstract and ¶ 88). In this process, the page creator module evolves a genome that can be represented as a tree structure in which the root 148 represents the entire photo album, the intermediate nodes 150-156 represent individual pages of the album, the intermediate nodes 158-164 represent subgroups within the individual pages, and the leaf nodes 166 represent the images (see FIG. 8 and ¶ 89). FIG. 8 shows a tree structure representation of the genome of the page creator module (see ¶ 21). The positions of the leaf nodes 166 within the tree structure shown in FIG. 8 assign the images to respective pages and to respective subgroups within the pages so that a fitness function score for each set of evolved image /page assignments can be determined (see, e.g., ¶¶ 97, 105-107, 110); the leaf node positions do not correspond to relative positions of the associated images within the area (i.e., album page) within which the images will be arranged. The page creator module transfers the image/page assignment information to the image placement module; all other information, including the assignment of images to subgroup information, is discarded (see, e.g., page 9, Table 3, which shows that the parameters for the image placement module do not include any information relating to the assignment of images to subgroups).

Geigel's image placement module uses a second genetic evolution algorithm to generate genetic structures of page layouts for images that are assigned to a given page (see abstract and ¶ 119). In accordance with this algorithm, the image placement module determines the absolute positions of the images on each of the album pages be evolving a genome that defines a respective layout of all of the images that have been assigned to the page by the page creator module (see ¶ 126: "The complete genome is comprised of the image positions of all images to be placed."). For each of the album pages, the genome that has the highest fitness score is selected as the image layout for that album page (see ¶¶ 145, 153, 155, and FIGS. 23-26).

2. Claim 1

a. Introduction

Independent claim 1 recites:

> 1. A method for arranging a set of objects within an area, comprising:
>
> initiating a first current binary tree comprising a leaf node;
>
> associating a first object selected from the set with the leaf node;
>
> establishing candidate binary trees, wherein each of the candidate binary trees comprises the current binary tree and a respective leaf node associated with another object selected from the set, and locations of the leaf nodes within each of the candidate binary trees correspond to relative positions of the associated objects within the area;
>
> computing a respective score for each of the candidate binary trees
>
> selecting one of the candidate binary trees as the current binary tree based on the computed scores;
>
> repeating the establishing, the computing, and the selecting until the current binary tree includes all the objects in the set; and
>
> after the repeating, arranging the objects within the area in accordance with the locations of the leaf nodes within the current binary tree.

As explained in detail below, the rejection of independent claim 1 under 35 U.S.C. § 102(b) over Burt should be withdrawn because Geigel neither expressly nor inherently discloses each and every element of the invention defined by the claim.

b. The Examiner's position

In support of the rejection of claim 1, the Examiner has taken the following position (see pages 3-4 of the final Office action):

- Geigel discloses the "initiating" element of claim 1in ¶¶ 69 and 89;

- Geigel discloses the "establishing" element of claim 1in ¶¶ 57-60, 64, 69, 70, and 97, and FIGS. 3, 8, and 9;
- Geigel discloses the "computing" element of claim 1 in ¶¶ 77 and 145-150;
- Geigel discloses the "repeating" element of claim 1 in ¶¶ 64 and 77; and
- Burt discloses the "arranging" element of claim 1 in FIGS. 3, 4, 9, and ¶¶ 77-79.

3. Appellant's rebuttal: Geigel does not disclose each and every element of the invention defined in claim 1

a. Geigel does not disclose the "establishing" element of claim 1

Contrary to the Examiner's position, Geigel does not disclose "establishing candidate binary trees, wherein each of the candidate binary trees comprises the current binary tree and a respective leaf node associated with another object selected from the set, and locations of the leaf nodes within each of the candidate binary trees correspond to relative positions of the associated objects within the area," as recited in claim 1.

Geigel's page creator module assigns images in a collection to album pages based on a first genetic evolution algorithm. In this process, the page creator module evolves a genome that can be represented as the tree structure shown in FIG. 8 in which the root 148 represents the entire photo album, the intermediate nodes 150-156 represent individual pages of the album, the intermediate nodes 158-164 represent subgroups within the individual pages, and the leaf nodes 166 represent the images (see FIG. 8 and ¶ 89). However, the positions of the leaf nodes within the tree structure shown in FIG. 8 do not correspond to relative positions of the associated images within the area (i.e., album page) within which the images will be arranged. Instead, the positions of the leaf nodes 166 assign the images to respective pages and to respective subgroups within the pages for the sole purpose of assigning images to pages (see, e.g., ¶¶ 97 and 115). Once the images have been assigned to pages, the subgrouping information is discarded; it is not transmitted to the image placement module for use in determining the positions of the images on the album pages (see, e.g., page 9, Table 3, which shows that the image placement module does not use any parameters that relate to the assignment of images to subgroups by the page creator module).

Geigel's image placement module uses a second genetic evolution algorithm to generate genetic structures of page layouts for images that are assigned to a given page. In

accordance with this algorithm, the image placement module determines the absolute positions of the images on each of the album pages by evolving a genome of the type shown in FIG. 17. This genome consists of a set of four positioning parameters (i.e., x-position, y-position, scaling, and rotation), which determine the absolute positions of the images to be placed on a given album page (see ¶ 126). However, neither the positioning parameters nor the genome of which they are a part constitute leaf nodes of a binary tree and, therefore, they cannot possibly have locations within a binary tree that correspond to relative positions of the associated objects within the area.

In response to these points, the Examiner has stated that (see page 7 of the final Office action dated Feb. 7, 2008; emphasis added):

> R1. Examiner does not agree. With the broadest interpretation of the current claim language and not reading into the specification as the Applicant implies the Examiner believes that Geigel does in fact teach that the location of the leaf nodes within each of the candidate binary trees correspond to relative positions of the associated objects within the area. For example please look at figure 9 and of course corresponding text relating to figure 9 (par.89). Geigel makes it clear of the structure of the binary tree and where the images are stored at in figure 8; in addition Geigel explains displaying the images and the output of the images in the end result in figure 9. As explained and depicted items 174,176, and 178 are subgroups of a page 172 each of the subgroups contain two images 182, 184 and 186 when processed through the system the end result is that of 172 (PAGE 2) depicted in figure 9. The Examiner poses this scenario which is completely within the scope of Geigel that if one of the leaf nodes of 186 was the child of subgroup 174 (keeping in mind that 174 is an event an grouping of events along with image analysis of the details of the image determines the location of the picture) and one of the leaf nodes of 182 was the child of subgroup 178 then the position (location) of the images from one of the nodes from 182 and 186 will be in different locations then what is depicted in figure 9 (PAGE 2), thus leading to the undeniable fact that Geigel fully supports establishing candidate binary trees in which the "location of the leaf nodes within each of the candidate binary trees correspond to relative positions of the associated objects within the area. Note paragraphs 59-60, 64, 89 and 97 for further support of the ideas expressed above. For instance note that an Event is a page, a sub event is a subgroup or pages and an event image is an image in a subgroup. In paragraph 97 Geigel points out "an image belonging to a sub

> event is grouped on the same page". <u>The placement of a leaf node in a tree that was established (created by the system for purposes of data organization) clearly indicates the effectiveness of the end result in such that an images location on a page in an album can be solely determined by the placement of the leaf node inside of the binary tree.</u>

The Examiner's response, however, is premised on various misunderstandings of Geigel's disclosure.

For example, the locations of the leaf nodes in the tree structure shown on the left half of FIG. 9 correspond to the subgroups to which they have been assigned by the page creator module. The locations of the leaf nodes within the tree structure, however, do not correspond to relative positions of the associated objects <u>within the area</u>, as recited in claim 1. That is, the assignment of images 182 to subgroup 174, images 184 to subgroup 176, and images 186 to subgroup 178 does not specify the page layouts of the images on the exemplary Pages 1 and 2 shown in FIG. 9. Instead, these page layouts correspond to "a possible layout solution" (¶ 89) that is determined by the image placement module (see, e.g., abstract, ¶¶ 88, 119). The assignment of images to subgroups is used only for the purpose of scoring each of the image/page assignments in terms of unity (see, e.g., ¶¶ 97 and 115). Once the images have been assigned to pages, the subgrouping information is discarded; it is not transmitted to the image placement mode for use in determining the positions of the images on the album pages (see, e.g., page 9, Table 3, which shows that the parameters used by the image placement module do not include any information relating to the assignment of images to subgroups).

In addition, ¶¶ 59-60, 64, 89 and 97 do not support the Examiner's position.

Paragraph 59 reads as follows:

> [0059] The overall function of the Page Layout System 124 is straightforward. Given a set of images to be placed in an album, a page layout algorithm must distribute the images amongst a set of pages and then layout the images on each individual page. Working within the framework of the AAS, the following information is available to the page layout system 124 on an image by image basis:
>
> [0060] Event Clustering—Images are grouped by event and sub-event

Contrary to the Examiner's position, this paragraph does not disclose anything whatsoever about establishing binary trees (candidate or otherwise) in which the "locations of the leaf nodes within each of the candidate binary trees correspond to relative positions of the associated objects within the area," as recited in claim 1. Instead, this paragraph merely states that the page layout system 124 distributes a set of images among a set of pages and then lays out the images on each of the album pages. As explained above, information relating to the grouping of images by event/sub-event is used solely for the purpose of assigning the images to pages of the album. After the images have been assigned to pages, the grouping information is discarded; it is not used by the image placement module in the process of determining the positions of the images on the album pages.

Paragraph 64 reads as follows:

> [0064] The page layout system 124 performs two separate, yet equally important tasks. Page creation 126, given a set of images, the system distributes these images amongst a set of album pages, such that each image is assigned a page upon which the image will appear. And, image placement 132, once the images have been assigned to pages, each individual page is laid out by positioning the images assigned to it. Therefore, for each image, placement, rotation, and scaling of the image on the page are assigned.

Contrary to the Examiner's position, this paragraph does not disclose anything whatsoever about establishing binary trees (candidate or otherwise) in which the "locations of the leaf nodes within each of the candidate binary trees correspond to relative positions of the associated objects within the area," as recited in claim 1. Instead, this paragraph merely states that the page layout system 124 assigns images to pages of an album and then determines the positions of the images assigned to each of the album pages.

Paragraph 89 reads as follows:

> [0089] The genome for the page creator module makes use of a tree structure as illustrated in FIG. 8. The root of the tree, node 148, represents the entire photo album. Interior nodes, 150, 152, 154, 156, 158, 160, 162, and 164 represent a structure of hierarchical visual groupings of images, which, in turn, are represented by the leaf nodes 166. However, it is equally suitable to replace this tree based encoding with a simpler data structure based on arrays. The first layer of the tree 148 indicates the partitioning of the album into pages. Nodes below

> the first layer represent visual groupings within pages 150, 152, 154, 156. The notion of this hierarchical visual grouping is illustrated in FIG. 9, which shows the encoding of an album with two pages and a possible layout solution that maintains the visual grouping relationships. The album is represented by node 168. It comprises two pages 170 and a72. Page One has three images 180. Page Two 172 has three subgroups of images 174, 176, and 178. Each of these subgroups has two images. Subgroup 174 having images 182, subgroup 176 having images 184, and subgroup 187 having images 186.

This paragraph does not disclose anything whatsoever about establishing binary trees (candidate or otherwise) in which the "locations of the leaf nodes within each of the candidate binary trees correspond to relative positions of the associated objects within the area," as recited in claim 1. The positions of the leaf nodes 166 within the tree structure shown in FIG. 8 do not correspond to relative positions of the associated images within the album page within which the images will be arranged. Instead, the positions of the leaf nodes 166 only assign the images to respective pages and to respective subgroups within the pages. The assignment of images to subgroups is used only for the purpose of scoring each of the image/page assignments in terms of unity (see, e.g., ¶¶ 97 and 115). Once the images have been assigned to pages, the subgrouping information is discarded; it is not transmitted to the image placement mode for use in determining the positions of the images on the album pages (see, e.g., page 9, Table 3, which shows that the parameters used by the image placement module do not include any information relating to the assignment of images to subgroups).

Paragraphs 93 and 97 read as follows:

> [0093] Evaluation Criteria—Solutions for the Page Creator Module are evaluated on the following four criteria:
>
> ...
>
> [0097] Unity—an evaluation of whether images belonging to the same event and/or sub-event are grouped on the same or subsequent pages.

Contrary to the Examiner's position, paragraph 97 does not disclose anything whatsoever about establishing binary trees (candidate or otherwise) in which the "locations of the leaf nodes within each of the candidate binary trees correspond to relative positions of the associated objects within the area," as recited in claim 1. Instead, paragraph 97 merely

indicates that the page creator module distributes a set of images among a set of pages based on evaluation criteria that includes an evaluation of whether images belonging to the same event and/or sub-event are grouped on the same or subsequent pages. As explained above, the grouping of images based on event/sub-event is used solely for the purpose of assigning the images to pages of the album. After the images have been assigned to pages, the grouping information is discarded; it is not used by the image placement module in the process of determining the positions of the images on the album pages (see, e.g., page 9, Table 3, which shows that the parameters used by the image placement module do not include any information relating to the assignment of images to subgroups).

For at least these reasons, the rejection of claim 1 under 35 U.S.C. § 102(b) over Geigel should be withdrawn.

b. <u>Geigel does not disclose the "repeating" element of claim 1</u>

As explained in the Amendment dated May 22, 2007, Geigel also does not disclose the process of "repeating the establishing, the computing, and the selecting until the current binary tree includes all the objects in the set," as recited in claim 1. Indeed, Geigel does not even hint that the page layouts generated by the image placement module are determined by repeatedly (i) establishing candidate binary trees comprising the current binary tree and a respective leaf node associated with <u>another</u> object selected from the set, (ii) computing respective scores for the candidate binary trees, and (iii) selecting one of the candidate binary trees as the current binary tree based on the computed scores. To the contrary, Geigel expressly teaches that for each album page the image placement module (i) evolves a genome that describes the positions of <u>all</u> the images assigned to the page to generate page layout genetic structures each of which defines a respective layout of <u>all</u> the images assigned to the page, and (ii) evaluates the page layout genetic structures, and (iii) selects the page layout genetic structure have the highest fitness score (see abstract; also see ¶ 126: "The complete genome is comprised of the image positions of all images to be placed.").

The Examiner's only reply to this point consists of the following statement (see pages 8-9 of the final Office action dated Feb. 7, 2008; emphasis added):

> R2. The Examiner does not agree. The repeating element of claim 1 is as follows: "repeating the establishing, the computing, and the selecting until the current binary tree

> includes all the objects in the set." Geigel describes his system as being able to process a set of images one at a time and not all at once hence "repeating" the process of establishing, computing and selecting. In paragraph 77 Geigel gives a summary of the overall architecture of the page layout system 124 in such that collection of images are inputted into the system and the system arranges these images in a fashionable manner that the user deems useful. This is done by computing the images by setting emphasis values on images so that images can be grouped and displayed together in the end result. The data structure choice to store the end result is a binary tree as depicted in figure 8 and the end result is depicted in at least figure 9 in such that one the system has processed the images and has established the tree the tree is then read by a viewer of the page layout system to present the page to the user as the user would deem useable means user preferences and the total over all processes from the page layout system. Thus the Examiner believes that Geigel does in fact teach repeating the establishing, the computing, and the selecting until the current binary tree includes all the objects in the set for at least the reasons stated above in R1 and R2 of this response.

The Examiner's observation that Geigel's system can process different sets of images at different times does not constitute a showing that Geigel discloses the "repeating" element of claim 1. Indeed, in the exemplary use case hypothesized by the Examiner each image set would be laid out on different sets of album pages and therefore the resulting layouts of the different image sets would not be on the same "area" (see the express language of claim 1).

The Examiner also has rested his position on the assertion that Geigel discloses the repeating element of claim 1 in paragraph 77 and FIGS. 8 and 9. The cited sections of Geigel's disclosure, however, do not support the Examiner's position.

Paragraph 77 reads as follows:

> [0077] FIG. 7 illustrates the overall architecture of a page layout system 124 according to an illustrative embodiment of the present invention. The system 124 takes as input a collection of images, which are clustered by events 116, 118, 120, 122 that are to be placed in an album. Being designed to work within the framework of an AAS, the system 124 receives the images that are sorted by event, duplicates and duds have been removed from the collection, and that each image has been assigned an emphasis value indicating the visual appeal of the image. The system produces a series of album pages 140, 142, 144, 146. There are two main system modules

> corresponding to the tasks outlined above. The Page Creator Module 126 is responsible for assigning each image to an album page. The page groupings created by this module are passed to the Image Placement Module 132, which positions the images on each individual page. Page layouts are specified in a textual form using an Extensible Markup Language (XML) format. Details about this format are available from: World Wide Web Consortium, Extensible Markup Language (XML) 1.0, http://www.w3.org/xml, February 1998, the contents of which are hereby incorporated by reference thereto. These textual descriptions are interpreted by a XML compiler 138 that creates composite images corresponding to each completed album page. Both the Page Creator Module 126 and Image Placement Module 132 make use genetic algorithms and consists of their own separate genetic engines, 128 and 136 respectively. More particularly, the Page Creator Module 126 comprises a genetic engine 128 and a page evaluation module 130, both of which are implemented in software on a processor as is understood by those of ordinary skill in the art. The Image Placement Module 132 comprises its own genetic engine 136 and a layout evaluation module 134, also implemented in software. In an illustrative embodiment, the present invention is implemented using Visual C++ 6.0 running under Windows 95/98/NT. Although porting to other platforms is readily feasible, as well as implementation in other programming languages and operation on other platforms, understood by those of ordinary skill in the art.

This paragraph, however, does not disclose for a single "area" repeating the establishing of candidate binary trees comprises the current binary tree and a respective leaf node associated with another object selected from the set, the computing of a respective score for each of the candidate binary trees, and the selecting of one of the candidate binary trees as the current binary tree based on the computed scores until the current binary tree includes all the objects in the set, as recited in claim 1. Instead, paragraph 77 discloses that the page layout system includes a page creator module 126 and an image placement module 132. The page creator module 126 assigns each image to an album page and passes the page assignments to the image placement module 132, which positions the images on each individual page.

FIGS. 8 and 9 also do not support the Examiner's position. As explained above, the locations of the leaf nodes in the tree structure shown on the left half of FIG. 9 correspond to the subgroups to which they have been assigned by the page creator model. The locations of the leaf nodes within the tree structure, however, do not correspond to relative positions of

the associated objects within the area, as recited in claim 1. That is, the assignment of images 182 to subgroup 174, images 184 to subgroup 176, and images 186 to subgroup 178 does not specify the page layouts of the images on the exemplary Pages 1 and 2 shown on the right side of FIG. 9. Instead, these page layouts correspond to "a possible layout solution" (¶ 89) that is determined by the image placement module (see, e.g., abstract, ¶¶ 88, 119). The assignment of images to subgroups is used only for the purpose of scoring each of the image/page assignments in terms of unity (see, e.g., ¶¶ 97 and 115). Once the images have been assigned to pages, the subgrouping information is discarded; it is not transmitted to the image placement mode for use in determining the positions of the images on the album pages (see, e.g., page 9, Table 3, which shows that the parameters used by the image placement module do not include any information relating to the assignment of images to subgroups).

For at least these additional reasons, the rejection of claim 1 under 35 U.S.C. § 102(b) over Geigel should be withdrawn.

c. Conclusion

For at least the reasons explained above, Geigel does not disclose each and every element of claim 1. Therefore the rejection of claim 1 under 35 U.S.C. § 102(b) over Geigel should be withdrawn.

3. Claims 2-7

a. Introduction

Each of claims 2-7 incorporates the elements of independent claim 1 and therefore is patentable over Geigel for at least the same reasons explained above.

Each of claims 3-5, and 7 also is patentable over Geigel for the following additional reasons.

b. Claim 3

Claim 3 depends from claim 2 and recites that the establishing of candidate binary trees comprises: removing a subtree of the current binary tree associated with a selected position within the current binary tree; inserting a new interior node into the current binary tree at the selected position; associating either a horizontal or a vertical partition of the area

with the new interior node; inserting into the binary tree a new leaf node emanating from the new interior node; associating the new leaf node with the other object selected from the set; and inserting the previously removed subtree back into the binary tree at the new interior node.

In support of the rejection of claim 3, the Examiner has taken the position that Geigel discloses "associating either a horizontal or a vertical partition of the area with the new interior node" in paragraph 57, lines 5-9. Paragraph 57, lines 1-10, reads as follows:

> [0057] The core processing of the AAS 2 includes several functions 26 that discriminate images and information for subsequent page layout. These include clustering of images by event 28, detection of dud images 30, detection of duplicate images 32, recognition of facial features and certain other objects 34, audio to text conversion 36, and video summarization 38. The reduced and refined image information is then coupled to a second group of core process functions 40 that further refine the image content information. ...

The section of paragraph 57 cited by the Examiner in support of the rejection of claim 3 does not disclose "associating either a horizontal or a vertical partition of the area with the new interior node," as recited in claim 3. Instead, this disclosure merely lists the following functions that are performed by the page layout system in the process of discriminating images and information for subsequent page layout: clustering of images by event 28, detection of dud images 30, detection of duplicate images 32, recognition of facial features and certain other objects 34, audio to text conversion 36, and video summarization 38. This disclosure does not even hint that the process of establishing a candidate tree comprises "associating either a horizontal or a vertical partition of the area with the new interior node," as recited in claim 3.

For at least this additional reason, the rejection of claim 3 under 35 U.S.C. § 102(b) over Geigel should be withdrawn.

c. Claim 4

Claim 4 depends from claim 3 and therefore is patentable over Geigel for at least the same reasons explained above in connection with claim 3.

d. Claim 5

Claim 5 depends from claim 2 and recites further comprising normalizing each of the candidate binary trees, wherein the normalizing comprises: for each of the interior nodes in the candidate binary tree, characterizing a respective bounding box for the objects included in the subtree rooted in the interior node; and for each of the objects, allocating a respective region of the area in accordance with the respective bounding box.

In support of the rejection of claim 5, the Examiner has taken the position that Geigel discloses "characterizing a respective bounding box for the objects included in the subtree rooted in the interior node" in FIG. 35 and paragraph 57, lines 1-2.

FIG. 35 shows an image placement example that illustrates the effects of emphasis (i.e., how well the scaling of images relate to image emphasis values) (see ¶¶ 144 and 162). FIG. 35 does not disclose anything whatsoever about the process of normalizing each of the candidate trees, much less does it disclose anything about characterizing a respective bounding box for the objects included in the subtree rooted in the interior node, as recited in claim 5.

Paragraph 57, lines 1-10, reads as follows:

> [0057] The core processing of the AAS 2 includes several functions 26 that discriminate images and information for subsequent page layout. These include clustering of images by event 28, detection of dud images 30, detection of duplicate images 32, recognition of facial features and certain other objects 34, audio to text conversion 36, and video summarization 38. The reduced and refined image information is then coupled to a second group of core process functions 40 that further refine the image content information. …

The section of paragraph 57 cited by the Examiner in support of the rejection of claim 5 does not disclose that the process of normalizing each of the candidate trees comprises "for each of the interior nodes in the candidate binary tree, characterizing a respective bounding box for the objects included in the subtree rooted in the interior node," as recited in claim 5. Instead, this disclosure merely lists the following functions that are performed by the page layout system in the process of discriminating images and information for subsequent page layout: clustering of images by event 28, detection of dud images 30, detection of duplicate images 32, recognition of facial features and certain other objects 34, audio to text conversion 36,

and video summarization 38. This disclosure does not disclose anything whatsoever about the process of normalizing each of the candidate trees, much less does it disclose anything about characterizing a respective bounding box for the objects included in the subtree rooted in the interior node, as recited in claim 5.

The Examiner also has taken the position that Geigel discloses "for each of the objects, allocating a respective region of the area in accordance with the respective bounding box" in FIG. 35 and paragraph 59, lines 1-4 and paragraph 70, lines 1-5.

FIG. 35 shows an image placement example that illustrates the effects of emphasis (i.e., how well the scaling of images relate to image emphasis values) (see ¶¶ 144 and 162). FIG. 35 does not disclose anything whatsoever about the process of normalizing each of the candidate trees, much less does it disclose anything about allocating a respective region of the area in accordance with the respective bounding box for each of the objects, as recited in claim 5.

Paragraph 59 reads as follows:

> [0059] The overall function of the Page Layout System 124 is straightforward. Given a set of images to be placed in an album, a page layout algorithm must distribute the images amongst a set of pages and then layout the images on each individual page. Working within the framework of the AAS, the following information is available to the page layout system 124 on an image by image basis:

Contrary to the Examiner's position, ¶ 59 does not disclose that the process of normalizing each of the candidate trees comprises allocating a respective region of the area in accordance with the respective bounding box for each of the objects, as recited in claim 5. Instead, this disclosure merely indicates that the page layout module includes a page layout algorithm that distributes images among a set of pages and then lays out the images on each of the pages.

Paragraph 70, lines 1-5, reads as follows:

> [0070] 2) Definition of Genetic Operators—New solutions are created via crossover and mutation of individuals from previous generations. Given a particular genome structure, the means for performing these operations must be defined. ...

Contrary to the Examiner's position, ¶ 70, lines 1-5, does not disclose that the process of normalizing each of the candidate trees comprises allocating a respective region of the area in

accordance with the respective bounding box for each of the objects, as recited in claim 5. Instead, this disclosure merely indicates that in a generic genetic search algorithms new genomes are created by well-known crossover and mutation operations.

For at least these additional reasons, the rejection of claim 5 under 35 U.S.C. § 102(b) over Geigel should be withdrawn.

e. Claim 7

Claim 7 depends from claim 1 and recites that the computing of a respective score for each of the candidate binary trees comprises assessing minimum and maximum object size values for all the objects in the area. Claim 7 also recites that the selecting of one of the candidate binary trees as the current binary tree based on the computed scores comprises selecting as the current binary tree the candidate binary tree having a greatest respective ratio of minimum area object size value divided by maximum area object size value.

In support of the rejection of claim 7, the Examiner has taken the position that in paragraph 90, lines 1-5, and paragraph 91, lines 2-7 Geigel discloses that the computing of a respective score for each of the candidate binary trees comprises assessing minimum and maximum object size values for all the objects in the area.

Paragraph 90, lines 1-7, reads as follows:

> [0090] Standard crossover and mutation operators for tree structures are used by the Page Creator Module. These operators are illustrated in FIG. 3 and FIG. 4, as discussed above. Note that the crossover operator may result in a solution that contains images appearing in multiple places within the tree or in solutions where particular images are omitted. ...

Contrary to the Examiner's position, ¶ 90, lines 1-5, does not disclose determining that the computing of a respective score for each of the candidate binary trees comprises assessing minimum and maximum object size values for all the objects in the area, as recited in claim 7. Instead, this disclosure merely indicates that the page creator module uses well-known crossover and mutation operations in the process of assigning images to pages.

Paragraph 91, lines 1-7, reads as follows:

> [0091] The determination of fitness used by the Page Creator Module is a combination of a number of factors. First, a

> solution is evaluated and scored using a number of different criteria. For each criterion, the score achieved by the solution is compared to the preference of the user as defined by the preference parameters supplied to the module. ...

Contrary to the Examiner's position, ¶ 91, lines 2-7, does not disclose determining that the computing of a respective score for each of the candidate binary trees comprises assessing minimum and maximum object size values for all the objects in the area, as recited in claim 7. Instead, this disclosure merely indicates that in the process of determining a fitness score for criterion the page creator module compares the score achieved by each solution to the user's preference.

The Examiner also has taken the position that in paragraph 110, lines 5-12 Geigel discloses that the selecting of one of the candidate binary trees as the current binary tree based on the computed scores comprises selecting as the current binary tree the candidate binary tree having a greatest respective ratio of minimum area object size value divided by maximum area object size value.

Paragraph 110, lines 5-12, reads as follows:

> ... Given a set of importance parameter values, the best possible score obtainable for a given run of the algorithm can be calculated by assuming the fitness score for each criterion to be perfect (i.e. equal to 1.0). The final fitness score is determined by scaling each of the actual fitness scores for each criteria by the corresponding importance parameter value, summing the results from all of the criteria and dividing this sum by the best possible fitness value obtainable. This final fitness score can be summarized mathematically as: ...

Contrary to the Examiner's position, ¶ 110, lines 5-12, does not disclose that the selecting of one of the candidate binary trees as the current binary tree based on the computed scores comprises selecting as the current binary tree the candidate binary tree having a greatest respective ratio of minimum area object size value divided by maximum area object size value, as recited in claim 7. Instead, this disclosure merely describes how a final fitness score is calculated.

For at least these additional reasons, the rejection of claim 5 under 35 U.S.C. § 102(b) over Geigel should be withdrawn.

4. Claim 8

Independent claim 8 recites features that essentially track the pertinent features of independent claim 1 discussed above. Therefore, claim 8 is patentable over Geigel for at least the same reasons explained above in connection with claim 1.

5. Claims 9-14

Each of claims 9-14 incorporates the features of independent claim 8 and therefore is patentable over Geigel for at least the same reasons explained above in connection with claim 8.

Claims 10-12 and 14 also are patentable over Geigel for at least the same additional reasons explained above in connection with claims 3-5 and 7, respectively.

6. Claim 15

Independent claim 15 recites features that essentially track the pertinent features of independent claim 1 discussed above. Therefore, claim 15 is patentable over Geigel for at least the same reasons explained above in connection with claim 1.

7. Claims 16-21

Each of claims 16-21 incorporates the features of independent claim 15 and therefore is patentable over Geigel for at least the same reasons explained above in connection with claim 15.

Claims 17-19 and 21 also are patentable over Geigel for at least the same additional reasons explained above in connection with claims 3-5 and 7, respectively.

VIII. Conclusion

For the reasons explained above, all of the pending claims are now in condition for allowance and should be allowed.

Charge any excess fees or apply any credits to Deposit Account No. 08-2025.

Respectfully submitted,

Date: June 9, 2008

Edouard Garcia
Reg. No. 38,461
Telephone No.: (650) 631-6591

Please direct all correspondence to:

Hewlett-Packard Company
Intellectual Property Administration
Legal Department, M/S 35
P.O. Box 272400
Fort Collins, CO 80528-9599

CLAIMS APPENDIX

The claims that are the subject of this Appeal are presented below.

Claim 1 (previously presented): A method for arranging a set of objects within an area, comprising:

initiating a first current binary tree comprising a leaf node;

associating a first object selected from the set with the leaf node;

establishing candidate binary trees, wherein each of the candidate binary trees comprises the current binary tree and a respective leaf node associated with another object selected from the set, and locations of the leaf nodes within each of the candidate binary trees correspond to relative positions of the associated objects within the area;

computing a respective score for each of the candidate binary trees

selecting one of the candidate binary trees as the current binary tree based on the computed scores;

repeating the establishing, the computing, and the selecting until the current binary tree includes all the objects in the set; and

after the repeating, arranging the objects within the area in accordance with the locations of the leaf nodes within the current binary tree.

Claim 2 (previously presented): The method of claim 1, wherein the current binary tree comprises:

at least one interior node; and

at least one leaf node emanating from the interior node;

wherein each subtree of the current binary tree comprises a respective position within the current binary tree and all interior nodes and leaf nodes emanating from the respective position; and

wherein each subtree of each of the candidate binary trees comprises a respective location within the candidate binary tree and all interior nodes and leaf nodes emanating from the respective location.

Claim 3 (previously presented): The method of claim 2, wherein the establishing comprises:

removing a subtree of the current binary tree associated with a selected position within the current binary tree;

inserting a new interior node into the current binary tree at the selected position;

associating either a horizontal or a vertical partition of the area with the new interior node;

inserting into the binary tree a new leaf node emanating from the new interior node;

associating the new leaf node with the other object selected from the set; and

inserting the previously removed subtree back into the binary tree at the new interior node.

Claim 4 (previously presented): The method of claim 3, wherein the selected position is selected from the group consisting of:

a leaf node in the current binary tree; and

an interior node in the current binary tree.

Claim 5 (previously presented): The method of claim 2, further comprising normalizing each of the candidate binary trees, wherein the normalizing comprises:

for each of the interior nodes in the candidate binary tree, characterizing a respective bounding box for the objects included in the subtree rooted in the interior node; and

for each of the objects, allocating a respective region of the area in accordance with the respective bounding box.

Claim 6 (previously presented): The method of claim 1, wherein the computing comprises determining respective fractions of the areas occupied by the objects in each of the candidate binary trees, and the selecting comprises selecting as the current binary tree the candidate binary tree having a greatest one of the fractions of the area occupied by the objects in the candidate binary tree.

Claim 7 (previously presented): The method of claim 1, wherein the computing comprises assessing minimum and maximum object size values for all the objects in the area,

and the selecting comprises selecting as the current binary tree the candidate binary tree having a greatest respective ratio of minimum area object size value divided by maximum area object size value.

Claim 8 (previously presented): A method for arranging a set of objects within an area, comprising:

establishing a tree structure;

associating a first object selected from the set with the tree structure to form a current candidate tree;

modifying the current candidate tree to form alternate candidate trees by associating a subsequent object selected from the set with different respective locations on the current candidate tree, wherein the respective locations correspond to relative positions of the associated objects within the area;

computing scores for each of the alternate candidate trees;

selecting one of the alternate candidate trees as the current candidate tree based on the computed scores;

repeating the modifying, the computing, and the selecting until all the objects in the set are associated with the current candidate tree; and

after the repeating, arranging all the objects in the set within the area in accordance with the locations in the current candidate tree associated with the objects.

Claim 9 (previously presented): The method of claim 8, wherein each of the tree structure, the candidate trees, and the alternate candidate trees comprises:

at least one interior node; and

at least one leaf node emanating from the interior node;

wherein each subtree of the current candidate tree comprises a respective location within the current candidate tree and all interior nodes and leaf nodes emanating from the respective location;

wherein each subtree of each of the alternate candidate trees comprises a respective spot within the alternate candidate tree and all interior nodes and leaf nodes emanating from the spot.

Claim 10 (previously presented): The method of claim 9, wherein the modifying comprises:

removing a subtree of the current candidate tree associated with one selected location within the current candidate tree;

inserting a new interior node into the current candidate tree at the selected location;

associating either a horizontal or a vertical partition of the area with the new interior node;

inserting into the current candidate tree a new leaf node emanating from the new node;

associating the new leaf node with the subsequent object selected from the set; and

inserting the previously removed subtree back into the current candidate tree at the new interior node.

Claim 11 (previously presented): The method of claim 10, wherein the selected location is selected from the group consisting of: a leaf node in the current candidate tree; and an interior node in the current candidate tree.

Claim 12 (previously presented): The method of claim 9, further comprising normalizing each of the alternate candidate trees, wherein for each of the alternate candidate trees the normalizing comprises:

for each of the interior nodes in the alternate candidate tree, characterizing a respective bounding box for the objects included in the subtree rooted in the interior node; and

for each of the objects, allocating a respective region of the area in accordance with the respective bounding box.

Claim 13 (previously presented): The method of claim 8, wherein the computing comprises determining respective fractions of the area occupied by the objects in each of the alternate candidate trees, and the selecting comprises selecting as the current candidate tree the alternate candidate tree having a greatest one of the fractions of the area occupied by the objects in the alternate candidate tree.

Claim 14 (previously presented): The method of claim 8, wherein the computing comprises assessing minimum and maximum object size values for all objects in the area for each alternate candidate tree, and the selecting comprises selecting as the current candidate tree the alternate candidate tree having a greatest respective ratio of minimum object size value divided by maximum object size value.

Claim 15 (previously presented): A method for arranging a set of objects within an area, comprising:

establishing a current candidate tree having at least one interior node, at least one leaf node connected to the interior node, and at least one object selected from the set associated with the leaf node;

modifying the current candidate tree to form alternate candidate trees by associating a subsequent object selected from the set with a different respective location on the current candidate tree, wherein the respective locations correspond to relative positions of the associated objects within the area;

computing scores for each of the alternate candidate trees;

selecting one of the alternate candidate trees as the current candidate tree based on the computed scores;

repeating the modifying, the computing, and the selecting until all the objects in the set are associated with the current candidate tree; and

after the repeating, arranging all the objects in the set within the area in accordance with the locations in the current candidate tree associated with the objects.

Claim 16 (previously presented): The method of claim 15, wherein each subtree of the current candidate tree comprises a respective location within the current candidate tree and all interior nodes and leaf nodes emanating from the respective location, and each subtree of each of the alternate candidate trees comprises a respective spot within the alternate candidate tree and all interior nodes and leaf nodes emanating from the respective spot.

Claim 17 (previously presented): The method of claim 16, wherein modifying the candidate tree comprises:

removing a subtree of the current candidate tree associated with one selected location within the current candidate tree;

inserting a new interior node into the current candidate tree at the selected location;

associating either a horizontal or a vertical partition of the area with the new interior node;

inserting into the candidate tree a new leaf node emanating from the new interior node;

associating the new leaf node with the subsequent object selected from the set; and

inserting the previously removed subtree back into the candidate tree at the new interior node.

Claim 18 (previously presented): The method of claim 17, wherein the selected location is selected from the group consisting of:

a leaf node in the current candidate tree; and

an interior node in current the candidate tree.

Claim 19 (previously presented): The method of claim 16, further comprising normalizing each of the alternate candidate trees, wherein for each of the alternate candidate trees the normalizing comprises:

for each of the interior nodes in the alternate candidate tree, characterizing a respective bounding box for the objects included in the subtree rooted in the interior node; and

for each of the objects, allocating a respective region of the area in accordance with the respective bounding box.

Claim 20 (previously presented): The method of claim 15, wherein the computing comprises determining respective fractions of the area occupied by the objects in each of the alternate candidate trees, and the selecting comprises selecting as the current candidate tree the alternate candidate tree having a greatest one of the fractions of the area occupied by the objects in the alternate candidate tree.

Claim 21 (previously presented): The method of claim 15, wherein the computing comprises assessing minimum and maximum object size values for all objects in the area for each of the alternate candidate trees, and the selecting comprises selecting as the current candidate tree the alternate candidate tree having a greatest respective ratio of minimum object size value divided by maximum object size value.

EVIDENCE APPENDIX

There is no evidence submitted pursuant to 37 CFR §§ 1.130, 1.131, or 1.132 or any other evidence entered by the Examiner and relied upon by Appellant in the pending appeal. Therefore, no copies are required under 37 CFR § 41.37(c)(1)(ix) in the pending appeal.

RELATED PROCEEDINGS APPENDIX

Appellant is not aware of any decisions rendered by a court or the Board that will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal. Therefore, no copies are required under 37 CFR § 41.37(c)(1)(x) in the pending appeal.